

Location, Location, Location: Exploring Amazon EC2 Spot Instance Pricing Across Geographical Regions - Extended Version

Nnamdi Ekwe-Ekwe and Adam Barker

School of Computer Science, University of St Andrews, UK

Email: {nnnee, adam.barker}@st-andrews.ac.uk

Abstract—Cloud computing is becoming an almost ubiquitous part of the computing landscape. For many companies today, moving their entire infrastructure and workloads to the cloud reduces complexity, time to deployment, and saves money. Spot Instances, a subset of Amazon’s cloud computing infrastructure (EC2), expands on this. They allow a user to bid on spare compute capacity in Amazon’s data centres at heavily discounted prices. If demand was ever to increase such that the user’s maximum bid is exceeded, their instance is terminated.

In this paper, we conduct one of the first detailed analyses of how location affects the overall cost of deployment of a spot instance. We analyse pricing data across all available Amazon Web Services regions for 60 days for a variety of spot instance types. We relate the data we find to the overall AWS region as well as to the Availability Zone within that region.

We conclude that location does play a critical role in spot instance pricing and also that pricing differs depending on the granularity of that location - from a more coarse-grained AWS region to a more fine-grained Availability Zone within a region. We relate the pricing differences we find to the price’s reliability, confirming whether one can be confident in the prices reported and subsequently, in the ensuing bids one makes.

We conclude by showing that it is possible to run workloads on Spot Instances achieving both a very low risk of termination as well as paying very low amounts per hour.

I. INTRODUCTION

Amazon EC2 allows developers to provision compute resources, configure them to their needs, as well as scale resources up or down depending on application requirements. EC2 is widely used by organisations and developers as it makes it easy for them to provision compute resources on-demand without having to invest considerable time and money into building the underlying infrastructure needed in a data center.

With EC2, developers have three major ways to pay what they provision. Developers can pay “On-Demand” which allows the developer to pay for what they use by the hour with no “long-term commitments” [2]. Developers can also pay via a “Reserved Instance” model allowing them to commit to using a set of instances for either a “1 or 3 year term” [2]. Amazon claims that this model provides a “significant discount (up to 75%) as compared to On-Demand pricing” [2]. Finally, developers can pay for resources using a “Spot Instance” model [2]. The “Spot Instance” model allows developers to bid on spare compute capacity giving them savings of up to “90% off the On-Demand price” [2].

Spot Instances are becoming increasingly useful for certain use cases. For example, users that suddenly have “urgent

computing needs for large amounts of additional capacity” [2] would find Spot Instances very useful. The user has access to both a sizeable amount of compute resources *and* at very low prices. The only caveat with this approach, however, is that the application that the user is running must be fault-tolerant due to the potential of an instance being terminated at any time.

The majority of research into Spot Instances has focused on modelling their pricing strategy as well as determining the best bid prices to make on an instance [4, 5, 10]. Papers such as [4, 10] have gone further - looking at how to find the lowest possible price for a user to spend, whilst maintaining the highest availability possible for the instance.

There is a distinct lack of research, however, which considers *where* (in terms of region and availability zone) a user can *reliably* deploy a spot instance in order to *minimise cost*. In order to address this gap, we make a number of core research contributions:

- We conduct one of the first detailed analyses of how location affects the overall cost of deployment of a spot instance.
- We analyse pricing data across all available Amazon Web Services regions for a variety of spot instance types. We relate the data we find to the overall AWS region as well as to the Availability Zone (AZ) within that region.
- For any pricing differences we find, we check whether those differences are *reliable* and as a result whether we can be confident in the ensuing bids we make.

We conclude that, 1) granularity of location has a significant impact on the overall price/price reliability of that instance; 2) the power of the instance type does not have as direct of an impact on price; some regions are universally substantially cheaper for certain instance types than others, while others are substantially more expensive.

The rest of this paper is structured as follows. Section II of this paper will examine what data we obtained from Amazon and its underlying structure. Section III will focus on the types of analyses we ran in order to explore the data. Section IV will give the results of our analyses. Section V will perform cross comparisons on our results to glean any interesting insights. Section VI will discuss related work. Finally Section VII will summarise and give our overall conclusions as well as discuss our future direction and work going forward.

II. EC2 SPOT PRICING DATA

Amazon provides spot price data to any Amazon user for a period of up to 90 days from when a request is made. This is done via its API endpoint [3]. This price data comes in JSON format in the form of a price point per time period.

For example:

```
{
  "Timestamp": "2017-06-25T00:20:56.000Z",
  "ProductDescription": "Linux/UNIX",
  "InstanceType": "d2.2xlarge",
  "SpotPrice": "0.177800",
  "AvailabilityZone": "eu-west-2a"
}
```

In the above example, the instance type (*d2.2xlarge*) was \$0.177800 (per hour) at 00:20:56 on the 25th of June 2017. The instance type was a Linux/UNIX instance - shown in the “ProductDescription” key. Finally, the Availability Zone in the EC2 region is *eu-west-2a*. We do not conduct any research related to the “ProductDescription” and so omit this column before we begin our analysis.

We used a data timeline of 60 days for our analysis. At time of carrying out this research, there were issues with retrieving the full 90 days worth of data from some zones. As a result, we worked with only 60 days worth of information. Additionally, not all Amazon regions provide a Spot Instance capability so our dataset comprised of data from only 4 AWS Regions - the EU, US, Asia Pacific and Canada Regions. These Amazon regions comprised of what, for the purposes of this paper, we’ll call *sub-regions*: EU - (*eu-central-1*, *eu-west-1*, *eu-west-2*), US (*us-east-1*, *us-east-2*, *us-west-1*, *us-west-2*), AP (*ap-southeast-1*, *ap-southeast-2*) and finally, Canada (*ca-central-1*). At time of running the analysis, there was a problem retrieving data from the *ap-northeast-1* sub-region and so those results have been omitted from the dataset.

There are 68 different instance types that can be launched on EC2. If we had obtained the data for all 68, this would have led to an extremely large dataset. We therefore decided to restrict the dataset based on the most popular instance types available: small, medium and large instance types [6]. Small instance types are not possible to deploy as a Spot Instance request so our final dataset consisted of medium and large instances. We then randomly picked a variety of instances (of varying compute power) from each of the AWS EC2 instance type categories - General Purpose, Compute Optimised, Memory Optimised and Storage Optimised [1].

Our final selection of data included *m3.medium*, *m4.large*, *c4.large*, *c3.large*, *r3.large*, *r4.large* and *i3.large* instance types. The total pricing data over the 60 days for all the above instances was 4,112,000 data points.

III. ANALYSES

We focused on exploring the pricing data in relation to i) the *instance type* and ii) the *availability zone*.

We ran three main analyses.

Instance Type	Mean±Standard Deviation(\$/h)
<i>c3.large</i>	0.077±0.077
<i>c4.large</i>	0.068±0.104
<i>i3.large</i>	0.181±0.496
<i>m3.large</i>	0.078±0.088
<i>m3.medium</i>	0.063±0.051
<i>m4.large</i>	0.052±0.054
<i>r3.large</i>	0.081±0.082
<i>r4.large</i>	0.080±0.055

TABLE I

MEAN AND STANDARD DEVIATION RESULTS OF ALL INSTANCE TYPES
IN THE EU OVER 60 DAYS (TO 3DP)

Instance Type	Mean±Standard Deviation(\$/h)
<i>c3.large</i>	0.074±0.104
<i>c4.large</i>	0.079±0.107
<i>i3.large</i>	0.081±0.115
<i>m3.large</i>	0.088±0.102
<i>m3.medium</i>	0.063±0.055
<i>m4.large</i>	0.084±0.118
<i>r3.large</i>	0.081±0.066
<i>r4.large</i>	0.090±0.127

TABLE II

MEAN AND STANDARD DEVIATION RESULTS OF ALL INSTANCE TYPES
IN THE US OVER 60 DAYS (TO 3DP)

Instance Type	Mean±Standard Deviation(\$/h)
<i>c3.large</i>	0.076±0.050
<i>c4.large</i>	0.096±0.139
<i>i3.large</i>	0.087±0.167
<i>m3.large</i>	0.085±0.056
<i>m3.medium</i>	0.063±0.047
<i>m4.large</i>	0.089±0.067
<i>r3.large</i>	0.088±0.058
<i>r4.large</i>	0.076±0.049

TABLE III

MEAN AND STANDARD DEVIATION RESULTS OF ALL INSTANCE TYPES
IN AP OVER 60 DAYS (TO 3DP)

Instance Type	Mean±Standard Deviation(\$/h)
<i>c4.large</i>	0.043±0.040
<i>i3.large</i>	0.072±0.045
<i>m4.large</i>	0.027±0.032
<i>r4.large</i>	0.029±0.033

TABLE IV

MEAN AND STANDARD DEVIATION RESULTS OF ALL INSTANCE TYPES
IN CA OVER 60 DAYS (TO 3DP)

- First, we ran an average price analysis of all instance types within an AWS region over the full 60 days.
- Second, we ran an average price analysis for each instance in our list of instance types for every AWS region.
- Finally, we ran a histogram analysis plotting the frequency of all price points in a particular AZ for each of the instances deployed in that zone.

We broke down the first two analyses by day of the week and by hour in the day - showing the mean price across both these time metrics. We also calculated the standard deviation and related the average price analysis to these in order to examine the price volatility. In this context, volatility means the propensity for the price to change across a time metric.

The lower the standard deviation, the lower the volatility. This meant that the data was closer to the mean and that we could be more confident in the mean price metrics reported.

Conversely, the higher the standard deviation, the less confident we could be in the price's reliability.

In the hour in the day analysis, we divided the day into 5 portions (00:00 to 05:00, 05:00-10:00, 10:00-15:00, 15:00-20:00 and 20:00-23:59). On the weekly analysis, we divided the week into 7 portions with 0 standing for Monday and 6 for Sunday.

IV. RESULTS

A. Average Price Analysis of All Instance Types within an AWS region

1) *EU*: At first glance, the average pricing for the instances seem to be mostly concentrated around the \$0.075 to \$0.080 mark. We can clearly see that *i3.large* instances are very expensive in comparison to the other instances with *m4.large* instances looking like the cheapest overall across the day.

When we related these prices to their standard deviations (Table I), we noticed some interesting metrics. In general across the EU, prices were very volatile (high levels of standard deviation). This was evident in some instance types more than in others - across the EU, *i3.large*, *c4.large* and *m3.large* instances had the three highest levels of standard deviation in the dataset. The three instances that showed the most reliable and stable pricing were the *r4.large*, *m4.large* and *m3.medium* instances (lowest levels of standard deviation). *m3.medium* instances in fact, showed the most consistent and reliable pricing throughout the day, followed by *r4.large* and *m4.large* instances.

Already, we can begin to see the impact location has on spot instance pricing - *r4.large* instances are some of the most powerful within the AWS offering - however it displays pricing similar to two of the least powerful instances in the dataset. This hourly breakdown offers us a very helpful view of pricing - we can see which hours of the day are cheaper to deploy an instance than in others. For example, we can see that at the beginning of the day (around midnight to 5am and at the tail end of the day - 8pm to midnight) we get some of the cheapest pricing available for *m4.large* instances. *m3.medium* instances show relatively consistent pricing throughout the day, a trend also similarly seen in *r4.large* instances types.

We also performed average pricing analysis over the week - giving us a higher level overview of price differences that we might not necessarily see on a daily basis. As seen in Figure 2(a), for *r4.large* and *m4.large* instances, the end of the week seems to be the best time to use them - offering the user the cheapest possible pricing. This, coupled with a low standard deviation allows us to conclude that these are reliable price metrics. *m3.medium* instances' pricing were also very consistent throughout the week. *i3.large* instances were clearly more expensive than all the other instances throughout the week. This, coupled with a very high standard deviation (also seen in *c4.large* and *m3.large* instance types)

shows the user that the EU is not the best place to deploy such instance types.

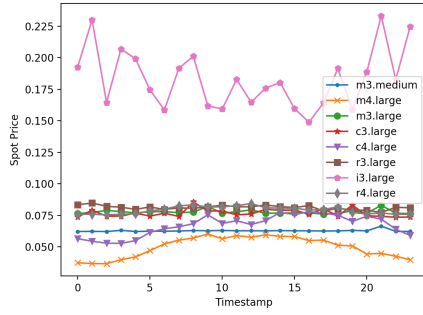
2) *US*: In the United States, we see significant and pronounced price volatility across nearly all the instance types. We see both very high levels of standard deviations (Table II) as well as high average price metrics throughout the day and throughout the week. Only two instance types (as seen in Figure 1(b)) display significantly low levels of standard deviations throughout the entire time period - *m3.medium* and *r3.large* instances. *m4.large* instance types display the most volatile pricing in the region, closely followed by *i3.large* and *m3.large* instance types. We see a few interesting things here. First, the type of instance clearly does not have as large an impact on price as location does. *m3* instance types are the least powerful in our dataset, yet we're seeing both very high and volatile pricing across the board. Conversely, we see reliable and low price metrics reported for *r3.large* instances, some of the most powerful in our dataset. We also see this replicated again in the week.

3) *Asia-Pacific*: The Asia-Pacific region, in comparison to the EU and US regions show both low price points coupled with low standard deviations (Table III). We see very low and reliable pricing in the *m3.medium*, *r4.large*, *c3.large* instances (Figure 1(c)). We again see the same pattern we saw earlier - more powerful instance types appearing at very low and reliable price points. In general, the more powerful the instance type, the more favourable the pricing in the Asia-Pacific region. *i3.large* instances once again display the most volatile pricing across the board, but for the most part, our previous conclusion holds. When we further examined some of the price metrics, we saw a sizeable difference in pricing with the Asia-Pacific region and the EU and US: *r3.large* and *r4.large* instances have max prices of \$0.1934 and \$0.14 per hour. In the EU, we see max prices 17x and 18x that for *r3.large* and *r4.large* instance types respectively, and 15x that in the US for both instance types. Across the week (Figure 2(c)), we see also see these low prices reflected.

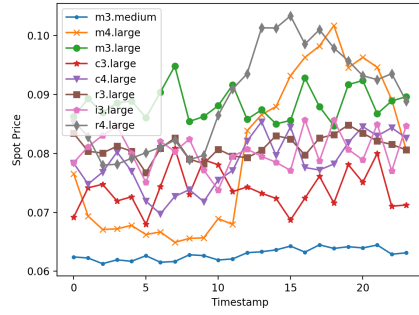
4) *Canada*: The Canada region hosts the least amount of instance types from our original selection - only the more powerful *c4.large*, *m4.large*, *i3.large* and *r4.large* instance types are present.

From these graphs (Figure 1(d) and Figure 2(d)) the Canada region exhibits some of the cheapest pricing on average per hour and across the week for the different instance types. For the *i3.large* instance type, we see mean prices at around \$0.07 per hour for the instance with the lowest standard deviations recorded across all four regions at \$0.045 (Table IV). In the Asia Pacific region, mean prices are at \$0.087 with standard deviations of \$0.167. In the US, \$0.081 with \$0.115 as the standard deviation and in the EU, \$0.181 with a standard deviation of \$0.496. From this we can clearly see that it would be much cheaper to deploy an *i3.large* instance in the Canada region than in any of the others. We can also apply the above analysis to all the other instance types.

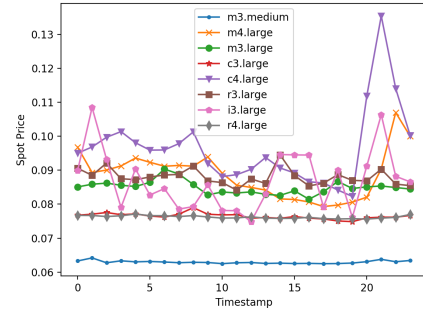
We can therefore conclude, that the more powerful the instance type, the cheaper and more reliable it is to deploy



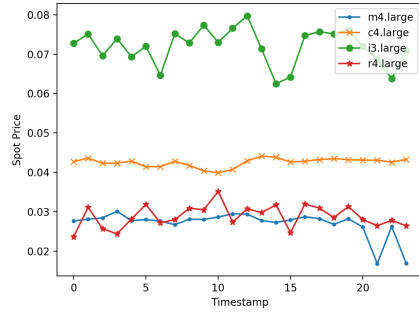
(a) Average Pricing Per Hour: EU



(b) Average Pricing Per Hour: US

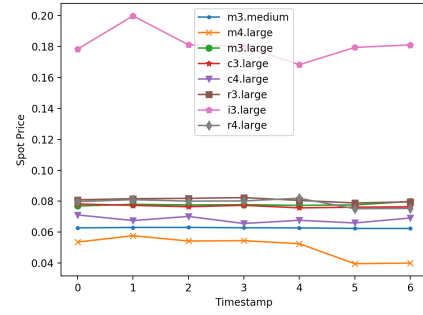


(c) Average Pricing Per Hour: Asia-Pacific

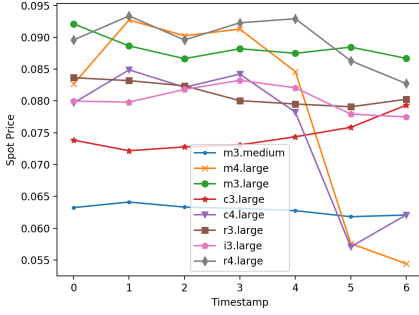


(d) Average Pricing Per Hour: Canada

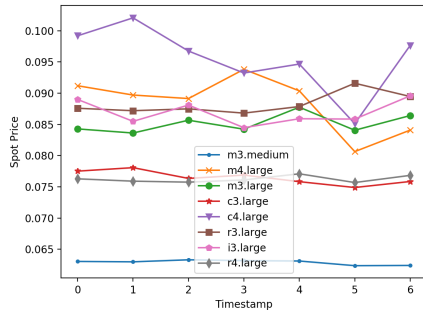
Fig. 1. Average Price Analysis of All Instances across all AWS regions: By Hour



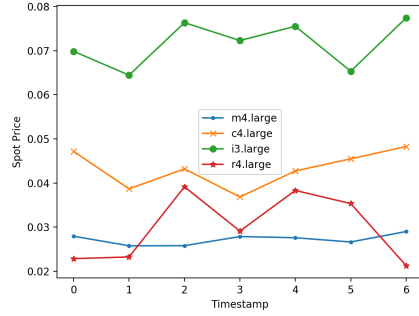
(a) Average Pricing By Day Of Week: EU



(b) Average Pricing By Day Of Week: US



(c) Average Pricing By Day Of Week: Asia-Pacific



(d) Average Pricing By Day Of Week: Canada

Fig. 2. Average Price Analysis of All Instances across all AWS regions: By Day of Week

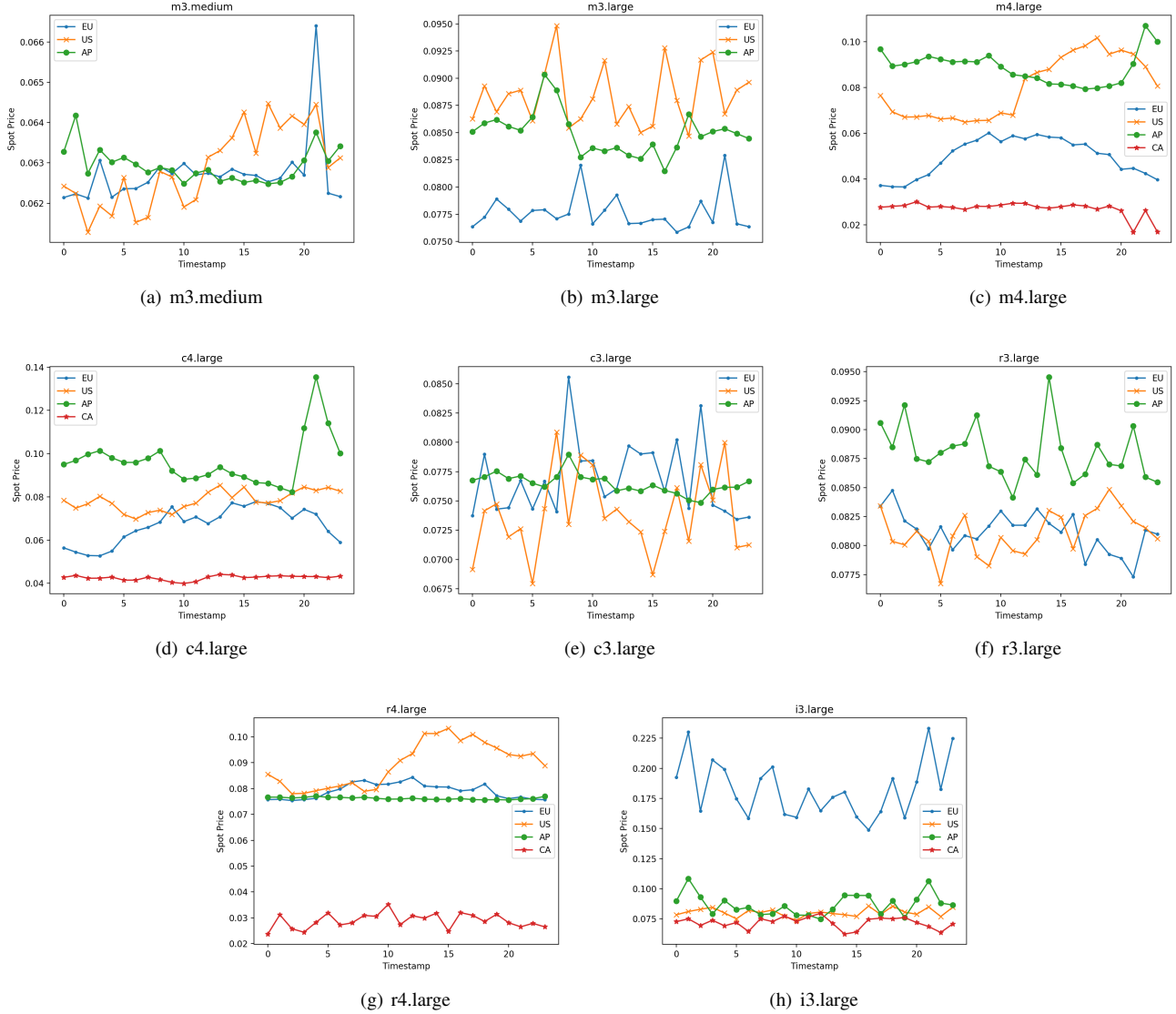


Fig. 3. Average Price Analysis of Each Instance for every AWS region: By Hour

in Canada.

B. Average Price Analysis of Each Instance for every AWS region

This analysis plotted every single instance type's pricing for every single AWS region. This allowed us to perform cross-regional comparisons and offered another view of the price data we had gathered. For some of the regions, some instance types were not deployed and as a result do not appear on some graphs.

We see some very clear patterns straight away. In any instance type that is deployed in Canada, we see both its lowest and most reliable price metrics. If we take a look at Figures 3 and 4, we see this very clearly. This also serves to reinforce the conclusions we came to earlier. The less powerful the instance, the more reliable it is to deploy in the EU or US - in the "m" instance types we generally see

cheaper pricing in those two regions.

For example, if one was to deploy an *m4.large* instance and had the choice between the AP region and the EU, one would obtain much more reliable pricing in the EU with a lower mean price and standard deviation than in AP. There are however, some anomalies - for example *m4.large* instance types being amongst the cheapest to run in the EU but the most expensive in the US. What one can see however, is that with these different price metrics we can begin to make informed decisions about where to deploy an instance. If (for instance) there were certain constraints that meant we could only deploy in a specific region over another, we can use this analysis to make the best choice for a particular use case.

C. Spot Price Histogram Price Frequency Analysis

1) *EU Region Analysis:* We can see that *m3.medium* instances are generally the cheapest to use amongst the

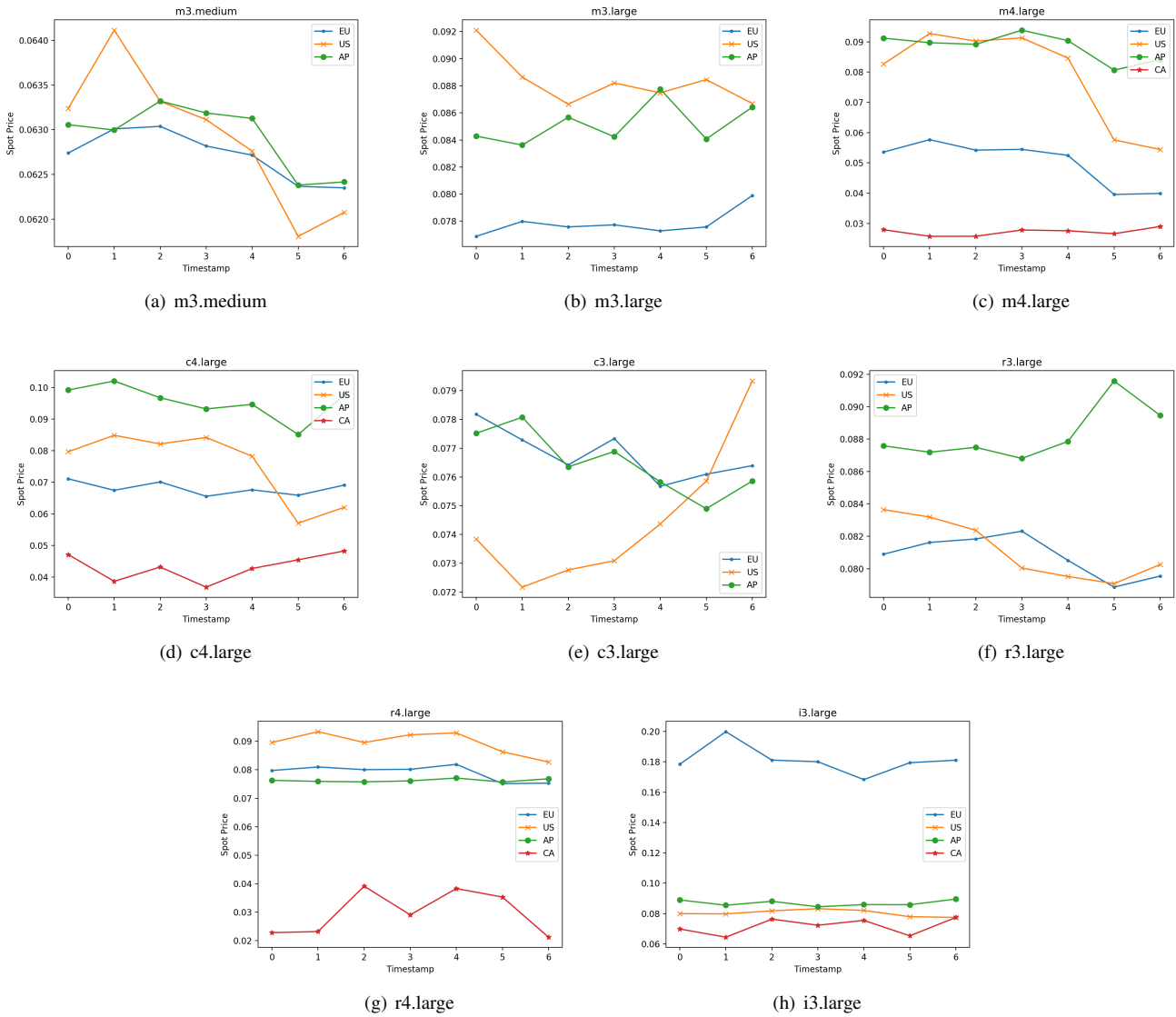


Fig. 4. Average Price Analysis of Each Instance for every AWS region: By Day of Week

availability zones (AZs) in which they have been deployed. However, in some AZs we do not see as high a frequency of data for *m3.medium* instances as some of the others. Note that the *eu-west-2a* AZ did not record any data for the *m3.medium* instances.

In the *eu-central-1a* AZ (Figure 5(a)), it was generally cheaper to deploy on a *c4.large* instance than it was on an *m3.large* instance. This is particularly interesting as *c4* instances are “optimized for compute-intensive workloads” [1]. *c4* instances are also more powerful than both *m3* and *m4* instances. This is a trend that was only really noticeable in the *eu-central-1a* AZ due to the amount of pricing data generated for *c4.large* instances. Looking across the other AZs however, *c4.large* instances had in several cases much cheaper pricing per hour than the *m3* and *m4* instance types.

Towards the other end of the scale, *r3* and *r4* instance types displayed some unexpected pricing trends. *r3* and *r4*

instance types are amongst the most powerful on AWS and are used to run “high performance databases...distributed web scale in-memory caches” [1] etc. However, in the *eu-west-1b* AZ (Figure 5(d)) we see *r4.large* instances displaying modal prices of around \$0.02 p/hour with a frequency of over 4000 and then of around \$0.12 p/hour (with a frequency of just over 3000). *r3.large* instances also generally fall into these above two “price brackets” (Figure 5(c) - *eu-west-1a* and Figure 5(e) - *eu-west-1c* AZs).

This is very interesting as it shows that 1) location of deployment is extremely important (specifically AZ) and 2) all these analyses must be taken together in context to find the best deployment area. A possible scenario - let's say a developer wanted to run a workload on a “r” instance type but data protection rules limited them to running this workload in Europe. From this analysis, the developer could run this workload quite comfortably in the *eu-west-1b* region and get

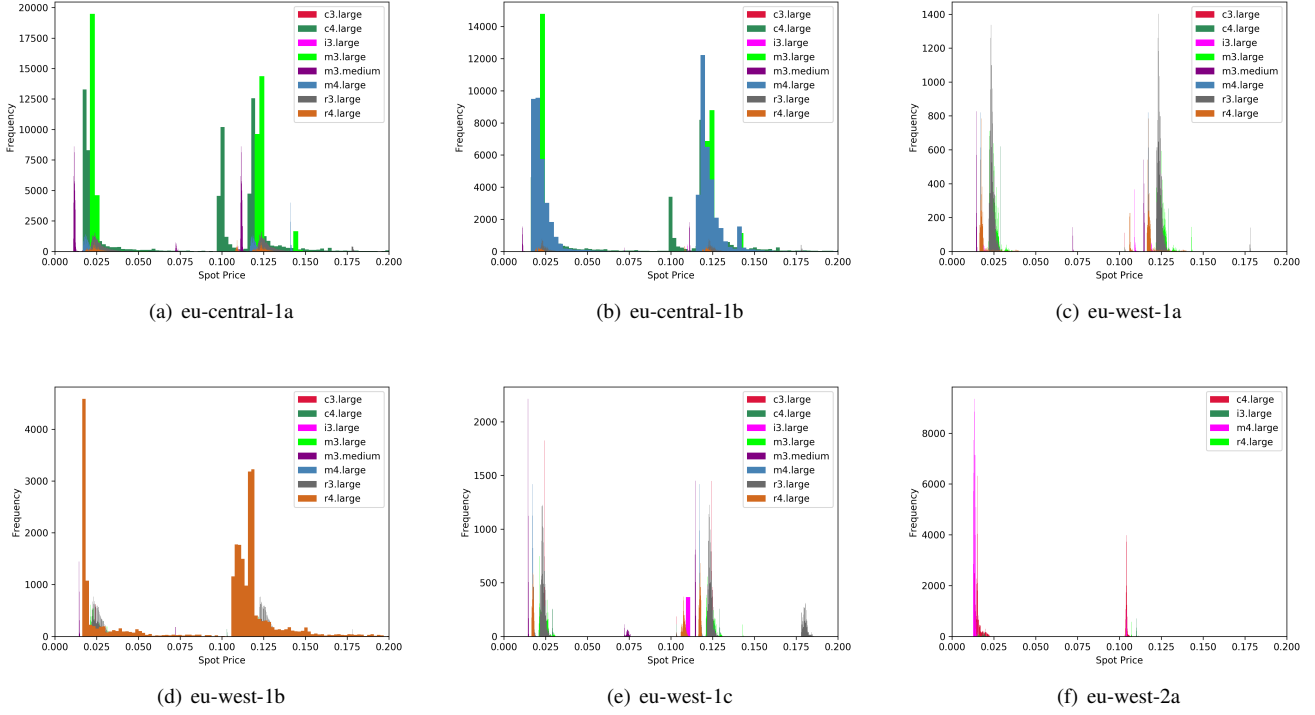


Fig. 5. Spot Price Histograms for all AZs within the EU

prices comparable to Canada or Asia Pacific regions.

2) *US Region Analysis:* *r4.large* instance types appear regularly on the left most side of the chart (i.e. the least expensive). In the *us-west-2b* AZ (Figure 6(h)), we see that *r4.large* instances are cheaper than *m4.large* instances with prices at around \$0.02 per hour even though *r4.large* instances are a lot more powerful. We see this again in the *us-east-1c* AZ (Figure 6(c)) with the mode pricing being around \$0.02-\$0.03 per hour as well as \$0.015-\$0.125.

m4.large instances follow their own distinct pattern across all the different AZs. Their two highest (modal) pricing points are either \$0.025 or \$0.125 per hour. We can see this most clearly in the *us-west-2a*, *2b* and *2c* AZs. In the *us-west-2a* AZ (Figure 6(g)) the two modal price points occur 6000 times with the next closest price point occurring around 1800 times.

In all AZs where they've been deployed, *m3.medium* instance types are consistently the cheapest instance type to use.

3) *AP Region Analysis:* *c3.large* instances show modal prices of either \$0.025 or \$0.125 an hour. The *c4.large* instance types generally fell within 3 modal pricing points in the *ap-southeast-2* regions - either \$0.03, \$0.10 or \$0.14 an hour.

The Asia Pacific region appears to be more popular with larger and more powerful instance types as that is where most of the data points are recorded for. Smaller and more General Purpose instance types have much smaller data frequencies although where they do occur, the data points generally

show cheap pricing. From this graph, we can clearly see advantages with deploying more powerful instance types (mainly c types) in this region than in the EU or US.

4) *Canada Region Analysis:* *m4.large* instance types are the cheapest instances to use in both the *ca-central-1a* and *ca-central-1b* AZs. In general prices across the zone are very inexpensive with prices not going past \$0.11 an hour mark. They also occur with high frequency. If one was deploying one of these 4 .large instances, one would deploy them here than any of the other regions.

Also, this graph has very few data points meaning that the region is not widely used a factor one could take advantage of to eliminate application “hotspots” as well as to get the lowest prices available.

V. CROSS-COMPARISONS

A. Average Price Analysis

The average price analysis over the day and week show us some interesting insights. We can see clear price differences in instance types as we go from region to region. In the Asia-Pacific and Canada regions we generally see much lower and more reliable pricing than in the EU and US regions. However, this is also instance type dependent. Overall, *m3.medium* instance types consistently show the most reliable pricing across all of the regions they are deployed in while *i3.large* instances show the most volatile. An interesting point to note is that *m3.medium* instances' mean prices are around the same point across all regions in which they are deployed - \$0.063 per hour.

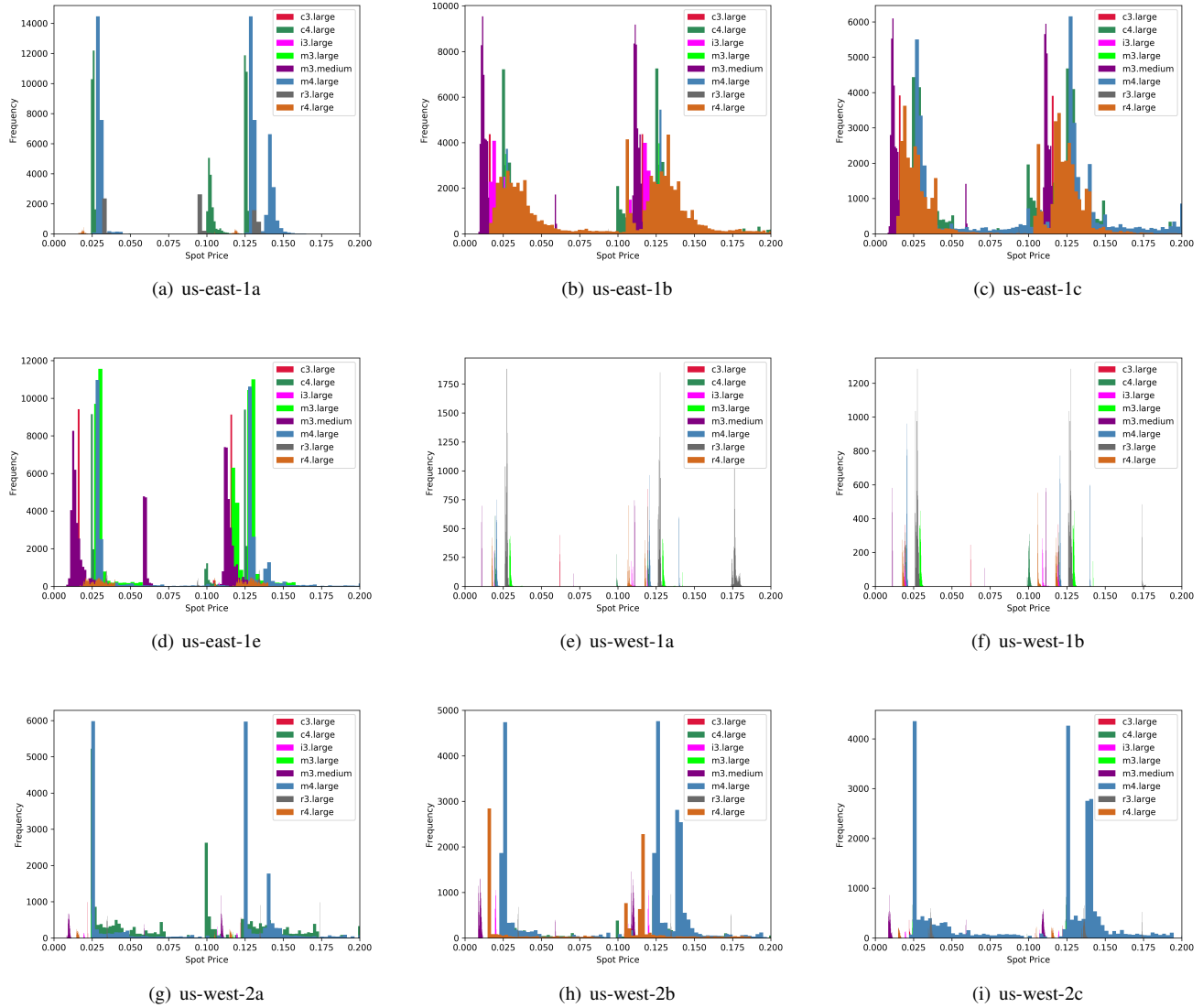


Fig. 6. Spot Price Histograms for all AZs within the US

We can use this analysis to examine which regions offer best value for the instance types we require. If one was to deploy an *i3.large* instance type, one of the most powerful in our list of instance types, we'd be looking at highly unstable pricing in the EU, US and Asia Pacific regions. However, in Canada, we'd be obtaining very cheap and reliable pricing per hour.

If one wanted to use a quite powerful *c3.large* instance, it would be much cheaper to deploy in the Asia Pacific region with a mean price of \$0.0764 per hour, a low standard deviation of 0.05, and max price of \$0.364 than in the EU or US. If one wanted to deploy a less powerful *m4.large* instance, however, one would much rather deploy it in the EU where the prices are on average \$0.052 p/hour with a standard deviation of 0.054 in comparison to \$0.089 p/hour and a standard deviation of 0.067 in the Asia Pacific region.

Based on all these analyses, there is a clear pattern identi-

fied. Overall, the more powerful the instance, the cheaper and more reliable the pricing one would get in Canada than in any of the other three regions. This is closely followed by the Asia Pacific region. On the other hand, the less powerful the instance the cheaper and more reliable (in terms of pricing) it is to deploy in the EU and US regions.

B. Spot Price Histograms

The Spot Price Histograms show us the frequency of prices for each instance type across the AZ. In the EU and US regions, we see a pattern - the "m" instances are generally the cheapest to run and appear in high frequencies towards the leftmost side of the graph.

However, in some AZs within these regions, we see that the more powerful instance types are among the cheapest to run - *r4.large* instances being an example. In the Asia Pacific and Canada regions, the *.large* instances present a

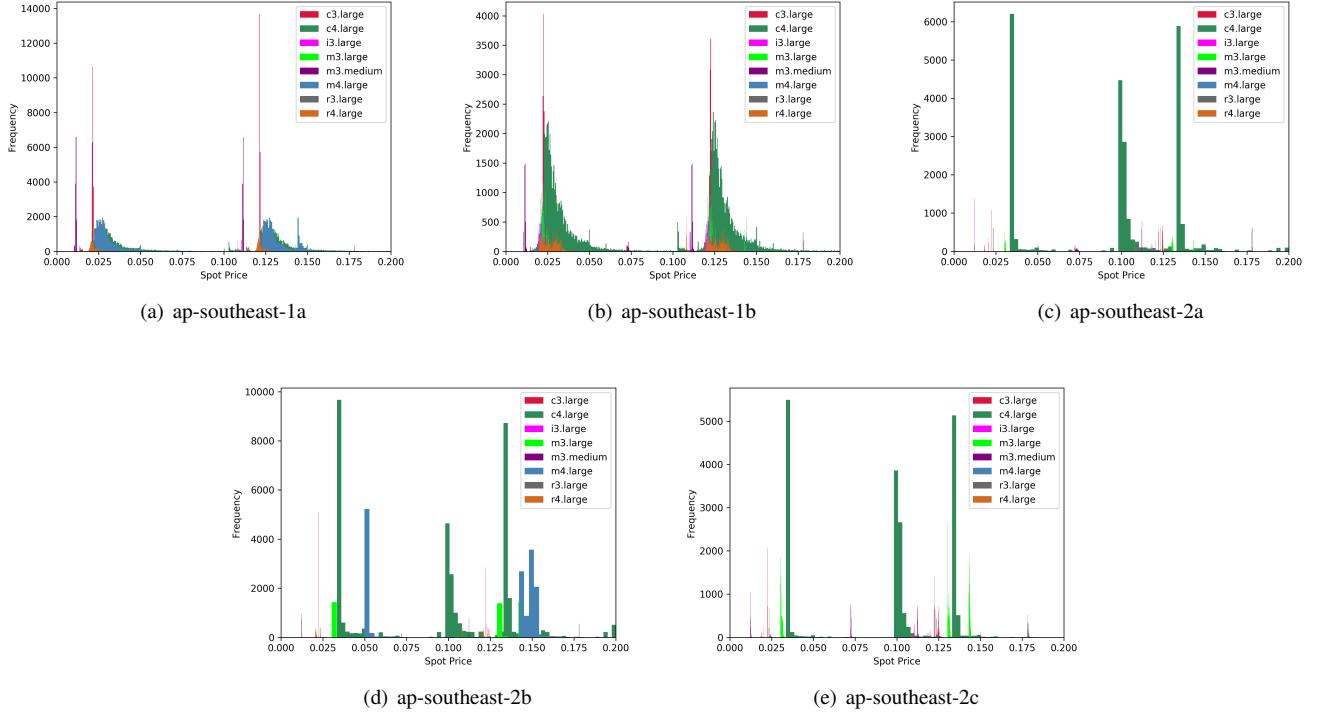


Fig. 7. Spot Price Histograms for all AZs within AP

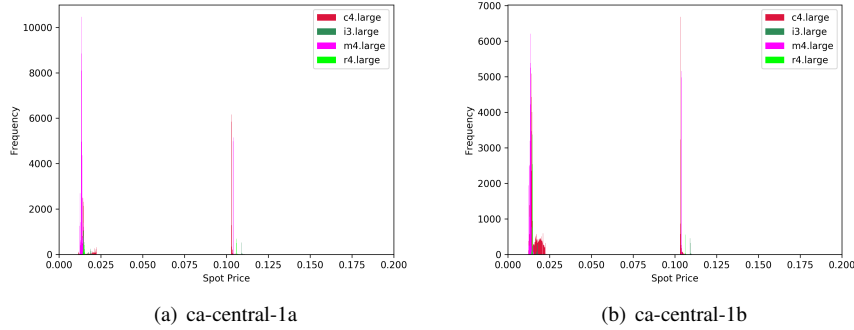


Fig. 8. Spot Price Histograms for all AZs within CA

double advantage of lower demand and lower prices.

These histograms can allow a developer to take a range of price points for a particular instance. A developer can then utilise the likelihood (modality) of those price points occurring and make a confident max bid price.

VI. RELATED WORK

There has been a significant body of research done into Amazon EC2 Spot Instances. However, most of the research done has been mainly focused on trying to determine how the pricing for the instances is generated. Bodies of work such as [4, 5, 10] look at modelling the pricing strategy for Spot Instances as well as determining the best bid prices to make on an instance. Papers [4, 10] try to find the lowest

price possible for a user to spend, whilst maintaining the highest availability possible for the instance.

Other bodies of research, more related to this paper, provide more meaningful insights. Wang et al., in [9] followed a more empirical approach to analysing Spot Instances, relating it to finding a more cost-effective procurement of compute resources for customers. They identified four key features a potential tenant of a Spot Instance should examine, “lifetime of an instance average spot price during lifetime simultaneous revocations, and startup delay”, created a quantitative model, and then used this model to develop “computationally-efficient predictors” of pricing [9].

Their conclusions revealed that “the evolution of spot price depended on a spatially coarse (e.g., data center or

availability zone wide) load metric” [9]. Their conclusions tie in fully with our own that the data center region or AZ had a significant impact on price. Approaching Spot Instances from an alternate angle, Li et al., recently in [11] performed a comparative investigation between Spot Instances and fixed-price instances using a Systematic Literature Review. They highlighted the fact that cloud consumers were concerned at “unpredictably frequent interruptions when using spot services” [11] - something that was putting them off utilising the spot market fully. The “academic community strongly advocate[d] the Cloud spot market” [11] but that that was still not being wholly reflected by consumers.

In recent times, however, this has started to change. Spotinst [8], a cloud company, launched in 2015 to “Reduce 50-80%” of cloud computing costs for companies on EC2 via the use of Spot Instances. It does this by utilising predictive algorithms to “to predict Spot behaviour, capacity trends, pricing, and interruption rate” [7], rebalancing whenever there is risk of an interruption, and falling back onto on-demand instances when Spot Instances are not available to use.

VII. CONCLUSIONS

In this paper, we have explored and gleaned fascinating insights from spot price data. We have seen the different dynamics that all play a role in affecting spot prices, including location (both on a coarse-grained regional level and on a finer grained availability zone level), type of instance, time of day, etc. We have analysed pricing volatility, determining when’s best to deploy an instance and whether we can be confident in the reliability of the average price point shown (via the standard deviation). We have also analysed modal price points for instance types in different AZs, giving us another viewpoint of spot price and allowing us to make confident predictions on potential bid prices.

We have shown how important context is in interpreting these analyses and that taken together, they paint a revealing picture about where best to deploy instances of varying power for the lowest price.

The general conclusions we can therefore draw are:

- I. The more powerful the instance, the cheaper it is in general to deploy in the Asia Pacific or Canada regions - the less powerful the instance, the EU and US regions.
- II. The Canada and Asia Pacific regions have a lot less demand, making it more attractive for developers (both from a pricing point of view) but also as a means of eliminating potential application “hotspots”.
- III. AZs have a significant impact on price - it is possible to deploy very powerful instances in a cheaper AZ located in an overall more expensive region. Figure 5(d) revealed to us how in the *eu-west-1b* AZ, we could deploy a very powerful *r4.large* instance for a very low price (\$0.02 and \$0.12 an hour).
- IV. There are some exceptions to these general conclusions one example is in the US where *m4.large* instances are amongst the most expensive to deploy while not being that powerful an instance type.

- V. Location plays a significant part in determining spot instance price, more than the instance type itself.

To conclude, spot Instances are very attractive as a means of using powerful compute power at very low prices. If a developer plans exactly where to deploy, examines all the pricing data available, and takes all the analyses performed together and in context, it is entirely possible to run workloads on Spot Instances at a very low risk of instance termination. Future work includes further analyses over longer periods of time, and building spot-price aware scheduling algorithms.

REFERENCES

- [1] Amazon. *Amazon EC2 Instance Types - Amazon Web Services (AWS)*. URL: <https://aws.amazon.com/ec2/instance-types/>.
- [2] Amazon. *Amazon EC2 Pricing - AWS*. URL: <https://aws.amazon.com/ec2/pricing/>.
- [3] Amazon. *describe-spot-price-history - AWS CLI 1.11.188 Command Reference*. URL: <https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-spot-pricehistory.html>.
- [4] Artur Andrzejak, Derrick Kondo, and Sangho Yi. “Decision Model for Cloud Computing under SLA Constraints”. In: *2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (2010). DOI: <https://doi.org/10.1109/MASCOTS.2010.34>.
- [5] Orna Agmon Ben-Yehuda et al. “Deconstructing Amazon EC2 Spot Instance Pricing.” In: *ACM Transactions on Economics and Computation* 1.3 (2013). DOI: <http://dx.doi.org/10.1145/2509413.2509416>.
- [6] Channel Futures. *2nd Watch Identifies Most Popular AWS Services, Instances — Channel Futures*. URL: <http://www.channelfutures.com/cloud-services/2nd-watch-identifies-most-popular-aws-services-instances>.
- [7] Spotinst. *Elastigroup - Reduce Cloud Compute Cost — Spotinst*. URL: <https://spotinst.com/workload-management/elastigroup/>.
- [8] Spotinst. *Spotinst: Your EC2 for 80% Less - The ultimate platform for running and managing Elastic Compute Cloud*. URL: <https://spotinst.com/>.
- [9] Cheng Wang, Qianlin Liang, and Bhuvan Ugaonkar. *Identification and Empirical Analysis of Amazon EC2 Spot Instance Features for Cost-Effective Tenant Procurement*. URL: <http://www.cse.psu.edu/research/publications/tech-reports/2016/CSE-16-006.pdf>.
- [10] Liang Zheng et al. “How to Bid the Cloud”. In: *SIGCOMM ’15 Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* 45.4 (2015), pp. 71–84. DOI: <http://dx.doi.org/10.1145/2829988.2787473>.
- [11] Li Zheng et al. “Spot pricing in the Cloud ecosystem: A comparative investigation”. In: *Journal of Systems and Software* 114 (2016), pp. 1–19. DOI: <https://doi.org/10.1016/j.jss.2015.10.042>.